Python科学计算模块

时间：10.14-11.15

本部分主要介绍Python中的科学计算模块：Numpy与Scipy的使用。现代科学都是建立在科学计算的发展基础之上。所以，如何存储数据，然后可以进行各种运算（例如：三角函数运算，指数运算，更高级别的代数运算等）也是数据科学的基础。

1、Numpy模块简介

NumPy（Numerical Python的简称）是Python数值计算最重要的基础包。大多数提供科学计算的包都是用NumPy的数组作为构建基础。

**功能：**

a.ndarray，一个具有矢量算术运算和复杂广播能力的快速且节省空间的多维数组。

b.用于对整组数据进行快速运算的标准数学函数（无需编写循环）。

c.用于读写磁盘数据的工具以及用于操作内存映射文件的工具。

d.线性代数、随机数生成以及傅里叶变换功能。

e.用于集成由C、C++、Fortran等语言编写的代码的A C API。

由于NumPy提供了一个简单易用的C API，因此很容易将数据传递给由低级语言编写的外部库，外部库也能以NumPy数组的形式将数据返回给Python。这个功能使Python成为一种包装C/C++/Fortran历史代码库的选择，并使被包装库拥有一个动态的、易用的接口。

NumPy本身并没有提供多么高级的数据分析功能，理解NumPy数组以及面向数组的计算将有助于你更加高效地使用诸如pandas之类的工具。

对于大部分数据分析应用而言，我最关注的功能主要集中在：

a.用于数据整理和清理、子集构造和过滤、转换等快速的矢量化数组运算。

b.常用的数组算法，如排序、唯一化、集合运算等。

c.高效的描述统计和数据聚合/摘要运算。

d.用于异构数据集的合并/连接运算的数据对齐和关系型数据运算。

e.将条件逻辑表述为数组表达式（而不是带有if-elif-else分支的循环）。

f.数据的分组运算（聚合、转换、函数应用等）。

虽然NumPy提供了通用的数值数据处理的计算基础，但大多数读者可能还是想将pandas作为统计和分析工作的基础，尤其是处理表格数据时。pandas还提供了一些NumPy所没有的领域特定的功能，如时间序列处理等。

NumPy之于数值计算特别重要的原因之一，是因为它可以高效处理大数组的数据。这是因为：NumPy是在一个连续的内存块中存储数据，独立于其他Python内置对象。NumPy的C语言编写的算法库可以操作内存，而不必进行类型检查或其它前期工作。比起Python的内置序列，NumPy数组使用的内存更少。

NumPy可以在整个数组上执行复杂的计算，而不需要Python的for循环。

**第一节 NumPy的ndarray：一种多维数组对象**

1、NumPy最重要的一个特点就是其N维数组对象（即ndarray），该对象是一个快速而灵活的大数据集容器。你可以利用这种数组对整块数据执行一些数学运算，其语法跟标量元素之间的运算一样。

要明白Python是如何利用与标量值类似的语法进行批次计算，先引入NumPy，然后生成一个包含随机数据的小数组;

print(type(data)): <class 'numpy.ndarray'>

注：

a. numpy.random.randn(d0, d1, …, dn)是从标准正态分布中返回一个或多个样本值。

b. numpy.random.rand(d0, d1, …, dn)的随机样本位于[0, 1)中;

可以对产生的data数组进行四则运算等计算；

ndarray是一个多维数组，其元素具有相同的基本类型。

1.可以通过属性ndim 来查看数组的维度

2.可以通过属性shape 来查看数组的各维度的长度

3.通过属性dtype查看数组元素的类型。

2、**创建ndarray**

⑴创建数组最简单的办法就是使用array函数。它接受一切序列型的对象（包括其他数组），然后产生一个新的含有传入数据的NumPy数组；

⑵嵌套序列（比如由一组等长列表组成的列表）将会被转换为一个多维数组；

⑶几种常用的产生特殊数组的方法

11. **zeros,zeros\_like**

**产生0数组的不同方法：**

**import numpy as np**

**np.zeros(5)**

**[0]\*5**

**[0 for \_ in range(5)]**

**np.array([0]\*5)**

**np.array([0 for \_ in range(5)])**

（1）zeros(shape, dtype=float, order=‘C’)，返回来一个给定形状和类型的用0填充的数组；

**参数**：

shape:数据尺寸

dtype:数据类型，可选参数，默认numpy.float64

（2）numpy.zeros\_like(*a*, *dtype=None*, *order='K'*, *subok=True*)形式如所给数据的样子，数值可以自己规定

**22. ones,ones\_like**

（2） [ones\_like](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ones_like.html#numpy.ones_like)

Return an array of ones with shape and type of input.

**33. empty, empty\_like**

（2）[empty\_like](https://docs.scipy.org/doc/numpy/reference/generated/numpy.empty_like.html#numpy.empty_like)

Return an empty array with shape and type of input.

**44. arange：range函数功能**

**55. full, full\_like：第一个参数是数组形式，第二个参数是数组值**

**66. eye：行列值三个参数；**

**Identity：方阵**

**77.** **dtype查看数组的格式，数据类型；**

可以使用stype方法明确地转换array的数据类型：

可以转换行列；

例如：np.arange(15).reshape([5,-1])

array([[ 0, 1, 2],

[ 3, 4, 5],

[ 6, 7, 8],

[ 9, 10, 11],

[12, 13, 14]])

**3、数组运算**

* 矢量运算：不用写循环
* 大小相等（shape 相同）的数组的数学运算，都是对应元素之间的运算
* 数组和标量的数学运算，将标量分别与对应元素运算

(1) arr>arr2 ## 产生布尔数组

array([[False, False, False],

[False, True, False]])

数组的shape不一样，也可以运算。这要用到数组的广播（broadcasting）,例如:

arr +np.arange(3)

array([0, 1, 2])

array([[1., 2., 3.],

[4., 5., 6.]])

array([[1., 3., 5.],

[4., 6., 8.]])

**4、数组的索引和切片**

* 用来选取部分数组
* 基本索引
* 布尔型索引
* Fancy 索引

(1)Reshape改变arange的数组形式

(2)np.T:转置处理

## 第二节 通用函数：快速的元素级数组函数

1、将函数作用在数组的对应的每一个元素上

* abs,sqrt,square,exp,log,log10,log2,isnan,cos,sin,modf
* power,maximum,minimum,fmax,fmin

（1）np.abs(data)

（2）dsqrt = np.sqrt(np.abs(data))

Dsqrt

（3）np.isnan(dsqrt)

array([[ True, True, True],

[False, False, False]])

（4）np.where(np.isnan(dsqrt))

（5）np.modf(data)

(array([[-0.26093275, -0.83193472, -0.78761312],

[ 0.59519751, 0.28725939, 0.35838248]]), array([[-0., -0., -0.],

[ 1., 0., 0.]]))

（6）frac\_part,whole\_part= np.modf(data)

frac\_part

array([[-0.26093275, -0.83193472, -0.78761312],

[ 0.59519751, 0.28725939, 0.35838248]])

## 第三节 利用数组进行数据处理

例1. 随机产生一个矩阵，然后将其中的正值换成“合格”，负值换成“不合格”

1.arr = np.random.randn(3,4)

arr

np.where(arr>0,"合格","不合格")

array([['合格', '合格', '合格', '合格'],

['合格', '合格', '不合格', '合格'],

['不合格', '合格', '不合格', '不合格']], dtype='<U3')

2. np.where(arr>0,"合格",arr)

array([['合格', '合格', '合格', '合格'],

['合格', '合格', '-0.5434522308859487', '合格'],

['-0.7494466763559184', '合格', '-0.6874599572892001',

'-0.2885220755741267']], dtype='<U32')

例2.给定一组观测（随机产生来模拟），求出各行、各列的和、均值、方差、最大值、最小值、最大值位置、最小值位置 找出数据的最大值位置

1. data.sum()

data.var()

data.max()

data.argmax()

np.unravel\_index(np.argmax(data), data.shape)

(4, 1)

data.shape

例3：判断data中是否有缺失值

1. np.isnan(data)

排序：data.sort()

**数组的去重／集合操作**

* unique() ##唯一元素提取
* intersect1d(x,y) ##交
* union1d(x,y) ## 并
* in1d(x,y) ##判断x中的元素是否在y中
* setdiff1d(x,y) ##差

## 第四节 用于数组的文件输入输出

真对numpy类型的变量，可以使用

* np.save
* np.load 进行快速的存储和读取

arr = np.arange(10)

np.save('myarr.npy',arr)

z = np.load('myarr.npy')

z

**第五节 线性代数**

二维数组其实就是数学中的矩阵，因此真对二维数组有许多矩阵的运算

* 矩阵乘法
* 矩阵分解
* 行列式
* 求逆矩阵

**第六节 随机数生成**

产生随机数，用于随机模拟

* 需要使用 numpy.random 模块

np.random.seed(1234) ### 设置随机数的种子

np.random.shuffle(x)

np.random.permutation(x)

**第七节 Scipy 模块简介**

SciPy是基于NumPy开发的高级模块，它提供了许多数学算法和函数的实现，用于解决科学计算中的一些标准问题。例如数值积分和微分方程求解，扩展的矩阵计算，最优化，概率分布和统计函数，甚至包括信号处理等。

作为标准科学计算程序库，SciPy类似于Matlab的工具箱，它是Python科学计算程序的核心包，它用于有效地计算NumPy矩阵，与NumPy矩阵协同工作。

* 线性代数
* 最优化
* 插值
* 统计